

## Code:

```
import java.util.HashSet;

import java.util.Set;

import java.util.Iterator;

class Tester {

    public static void main(String[] args) {

        // Creating HashSet

        Set<String> food = new HashSet<String>();

        // Checking if a HashSet is empty

        System.out.println("Is the set empty? : " + food.isEmpty());

        // Adding elements to the HashSet

        food.add("Pasta");

        food.add("Noodles");

        food.add("Sandwich");

        food.add("Pasta");

        food.add("Burger");

        food.add("Noodles");

        System.out.print("Set output without the duplicates: ");

        System.out.println(food);

        // Finding the size of the HashSet

        System.out.println("The number of food items in the set: " + food.size());

        // Checking if the HashSet contains the given element

        String foodItem = "Pasta";

        if (food.contains(foodItem))

            System.out.println(foodItem + " is already ordered");
```

```

else

    System.out.println(foodItem + " is not ordered");

// Removing an element from the HashSet
if(food.remove("Burger"))

    System.out.println("Output after removing Burger from the set:" + food);

// Traversing elements
Iterator<String> item = food.iterator();

while (item.hasNext())

    System.out.println(item.next());

// Removing all the elements from the HashSet
food.clear();

System.out.println("After clear() => " + food);

}

}

```

### Output:

```

Is the set empty? : true
Set output without the duplicates: [Sandwich, Burger, Pasta, Noodles]
The number of food items in the set: 4
Pasta is already ordered
Output after removing Burger from the set:[Sandwich, Pasta, Noodles]
Sandwich
Pasta
Noodles
After clear() => []

```